Indian Journal of Science and Technology, Vol 8(30), DOI: 10.17485/ijst/2015/v8i30/87232, November 2015
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Triple Data Encryption Standard Encryption Engine: A Hardware Approach

Mohd. Marufuzzaman, Noorfazila Kamal, Fazida Hanim Hashim and Mamun Bin Ibne Reaz*

Universiti Kebangsaan Malaysia, Department of Electrical, Electronic and Systems Engineering, Bangi - 43600, Selangor, Malaysia; mamun.reaz@gmail.com

Abstract

Cryptography is known as the standard means of rendering a communication private. This research work describes an approach to develop Triple Data Encryption Standard Encryption Engine in FPGA that can be used as a standard device in the secured communication system. The hardware design has been targeted to implement on Altera FLEX10K and FLEX10KE devices. By trading off between the processing time and the security matters the key size of the 3DES encryption engine has been set to 64-bit, which practically provides a considerable amount of security to the communication system. The 3DES encryption engine has made use of 239 units of Logic Cell (LC) with 199MHz. It has been verified that this 3DES encryption engine can perform the 64-bit operation in less than 22.38us.

Keywords:

1. Introduction

In the wake of advancement in computer technology and increasingly volatile information flow, we are faced with challenges of safeguarding information that is not meant for public knowledge. Cryptography is known as a way to keep certain information private by encryption and decryption operations[1]. Encryption is the transformation of data into cipher text whereas decryption is the reverse process of encryption where the cipher text is transformed back to the original information. The encryption and decryption of data is based on certain secret information called key[1,2].

Basically, cryptography can be divided into two types: symmetric and asymmetric. Symmetric encryption, also known as secret key cryptography, is a mechanism where both encryption and decryption use a single common key. As such, the sender and receiver need to agree on a specific key. Any interception during the transmission of the key will enable the important data to be read. Examples of symmetric encryption are DES, Triple DES and AES. Asymmetric encryption or public key cryptography employs two different keys, which are public key and private key. The public key is used for encryption whereas the private key is for decryption. The encrypting key is made public so that data can be enciphered by anyone who is in possession of the public key. However, the decryption of information can only be done by the owner of the private key. Examples for this encryption type are RSA, Diffie-Hellman, PGP and Elliptic curve.

Hardware implementation of Triple DES had been chosen based on its close relationship to DES[3]. With an increase in security and compatibility to DES software and hardware, Triple DES is a better choice compared to other algorithm[3,4]. Due to this reason, Triple DES will remain important in foreseeable future even though AES has been selected as a replacement for DES.

Triple DES (3DES), an improved version of DES provides a better security compared to DES[3]. This is due to a longer key length and more rounds of DES encryptions. DES only has a key length of 56 bits which is insufficient to resist any brute force attack today[5]. A key breaking machine cost less than $1 million can find a key in an average of 3.5 hours and the cost is estimated to drop by a factor of 5 every 10 years[5]. Even though 3DES is thrice slower than DES, however, if used properly, it can be as strong as 2304-bit public key algorithm due to its longer key length[3,4].

*Author for correspondence

Triple DES is more efficient compared to RSA and Elliptic curve (ECC). Due to its symmetric nature, 3DES is a better choice in encrypting bulk data and is therefore less expensive[2,6]. 3DES uses only 128 or 196 bits symmetric keys and has simpler algorithm. It is less computationally intensive and does not introduce much overhead. Thus, it requires relatively inexpensive hardware. Due to its much longer key length, RSA is not suitable to be used in mobile or wireless devices as these devices have underpowered processors[7]. ECC founded on 1985, on the other hand, is deemed as immature and it requires much research into its elliptic curve discrete logarithm problem. In ECC, certain curves can be solved in sub-exponential time, hence it can be broken easily if these curves are used. For RSA, factoring of small numbers is easy. As such, ignorance of these behaviors will lead to serious implications.

In comparison to AES, Triple DES is faster[8]. The limitation of AES exists in that the cipher and its inverse use different code and/or tables. As such, it does not have bidirectional architecture for encryption and decryption as that of Triple DES. The inverse cipher can only partially reuse the circuitry that implements the cipher, resulting larger hardware presumably.

As DES is the most widely used secret key encryption algorithm, Triple DES offers the advantage of compatibility with legacy DES software and hardware systems unlike those of RSA, MD5 etc[3]. This gives a cheaper solution in improving security of a certain system. Triple DES also inherits the advantage of DES of having simple bidirectional architecture for encryption and decryption[6].

Hardware realization of Triple DES has the advantage of being more secured and faster. Even though software implementation cryptography uses general-purpose processors that offer enough power to satisfy the needs of individual, hardware realization is the only way to achieve speeds more significant than general-purpose microprocessor[9]. This is important for commercial and communication purposes as the security related processing can consume up to 95% of a server's processing capacity. By using a dedicated hardware running encryption application, this will render parallel processing capabilities which can get more computing done within a stipulated period[1]. As hardware has only an input and an output, it can be important in the protection process against hacker. The encrypting and decrypting part will be done inside the hardware, resulting in impossibility of tampering the hardware without physical access to it[9]. Higher security can be provided with the tamper resistant features of hardware which does not leak any information under any form of stressing.

In contrast, software implementation is susceptible to small program like virus[9]. Virus can be used to imperceptibly collect information to be sent to the attacker. Besides that, scanning of memory freed by cryptographic program that finish execution will release information to the hacker. Plaintext or the cipher text obtained can lead to the discovery of key used by using linear attack[10].

Two major approaches of hardware implementation of encryption are Field Programmable Gate Arrays (FPGAs) or Application Specific Integrated Circuits (ASICs). FPGA is a more flexible device as it is possible to be reconfigured so that unused encryption schemes can be removed from the chip to save resources. It also offers lower cost for small scale production and a shorter development. This is because the final layout of ASIC has to be sent to very expensive fabrication. Due to the high development cost, any mistakes done during the development process will lead to higher cost. This results in a longer development time for ASIC. Besides that, FPGA based systems support bug fixes and new standards to be upgraded. As such, it can act as an inexpensive prototyping.

Hardware realization of Triple DES has a wide range of applications. It is especially important in the electronic and financial transactions such as E-Commerce and banking[6,11]. It can also be used for wireless local area network and Virtual Private Networks (VPN)[12]. Wireless connection's security had been compromised wherein data is available to anyone within the range of transmitting device, resulting in serious security violations and the need of powerful encryption[12]. VPNs are architectures to realize connections among different private network via a public network. As public network is insecure, cryptography provides a way to provide a secure communication channel between the private networks and the public network. Other applications are in the secure IP (IPSEC) and network management[11].

2. Methodology

Triple DES encrypts a block of 64 bits data using two or three unrelated 64 bits keys. The internal operation done on these data is similar to that of DES where the only difference is that DES consists of 16 iterations whereas Triple DES iterates three times that amount. Out of the 64 bits keys being used in DES, the effective key size is only 56 bits. The eighth bit is used for odd parity checking and is

Table 1. PC-1

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	21	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Table 2. Iterations schedule

n	1	2	3	4	5	6	7	8
left shift	1	1	2	2	2	2	2	2

n	9	10	11	12	13	14	15	16
left shift	1	2	2	2	2	2	2	1

Table 3. PC-2

14	7	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Table 4. IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

thus ignored. As such the total effective key size for Triple DES is 3 times 56 bits.

In a DES encryption operation, it can be divided into two stages involving the key and the data. During the first stage, permutation is done on the 64 bits key based on the following table.

After permutation, the 56 bits key will be split into left and right halves C_0 and D_0, having 28 bits each. With C_0 and D_0 defined, sixteen blocks C_n and D_n, where n=1,2,3,….,16 can be formed by left shifting C_{n-1} and D_{n-1} once or twice using the following schedule of table 2. C_n and D_n will then be concatenated before applying the following table on C_nD_n, resulting in 48 bits key.

In the second stage, there is an initial permutation IP on the 64 bits message data M in accordance to the following table. The permutated M, M' is then divided into left half L_0 and right half R_0, each having 32 bits.

It is followed by 16 iterations of operations, n = 1,2,3,….,16 using function f which operates on two blocks: data block of 32 bits and key block of 48 bits to produce a block of 32 bits.

$$L_n = R_{n-1} \quad (1)$$

$$R_n = L_{n-1} + f(R_{n-1}, K_n) \quad (2)$$

In each iteration, the right 32 bits of the previous iteration will be used as the left 32 bits of the current iteration. The right 32 bits in the current iteration will be provided by XORing the left 32 bits of the previous step with function *f*. To calculate *f* function, each block R_{n-1} will be expanded from 32 bits to 48 bits using the Expansion table E, resulting in $E(R_{n-1})$.

$E(R_{n-1})$ is then XORed with the block of key K_n, where n=1,2,3,….,16, resulting in $B_1B_2B_3B_4B_5B_6B_7B_8$, where B_i is a group of 6 bits. B_i will be used as the addresses in tables called "S Boxes". Each B_i gives an address in a different S box, S_i. The first and last bits of Bi represent a base 2 number in the decimal range of 0 to 3. This indicates the row number in the S Box. The middle 4 bits of B_i will give the decimal range 0 to 15, indicating the column number in the S Box as shown in Table 5.

$$K_n + E(R_{n-1}) = B_1B_2B_3B_4B_5B_6B_7B_8 \quad (3)$$

Function *f* can be obtained by doing a permutation on the S-Box output according to the table 6.

$$F = P(S_1(B_1)S_2(B_2) \ldots S_8(B_8)) \quad (4)$$

At the end of the sixteenth iteration, the order of the two blocks is reversed into 64-bit block before applying a final permutation as defined in the table 7.

Decryption in DES uses the same process as encryption operation, but with the order of the key being used in reverse order. This means that instead of K1-K16, K16 is applied first with K1 being applied last.

Encryption and decryption using Triple DES can be done by compounding the operation of DES encryption Ek(I) and decryption Dk(I) operations. Encryption oper-

Table 5. Expansion table

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Table 6. P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

ation in Triple DES can be defined by Ek3(Dk2(Ek1(I))) whereas the decryption operation is defined as Dk1(Ek2(Dk3(I))). The keys being used may vary according to 3 different keying options. The first keying option is that K1, K2 and K3 can be independent keys. The second keying option is that K1 and K3 can be the same key with K2 being the independent key. The last keying option is that K1, K2 and K3 are the same, giving a slower version of DES.

In terms of hardware implementation, the first step of designing with FPGA was to draw a block diagram of the overall system. The system is then described using VHDL and compiled by a VHDL compiler. Functional simulation is run to test the correctness of the hardware design. Once the functionality of the design is verified, the VHDL code is modified to make sure that it is synthesizable. A synthesis tool is run to generate logic for the design. After logic synthesis, a partitioning program is used to divide the logic circuit into pieces that will fit into the configurable logic blocks in a FPGA. This is followed by a place and route program that will arrange the logic blocks in proper places in the FPGA and then route the interconnections between the logic blocks. Finally, an assembler is run to generate bit pattern necessary to configure the FPGA. The bit pattern is downloaded into the internal configuration memory cells in the FPGA. Further testing can be performed to validate the operation of the FPGA. When the final system is built, the programming bit pattern is normally stored in an EPROM, which is loaded into the FPGA during power on.

Table 7. 64-bit block

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

The targeted FPGA device is basically Altera FLEX10KE device family. It is the embedded programmable logic devices (PLDs), providing system-on-a-programmable-chip (SOPC) integration in a single device. FLEX10KE family provides higher density which from 30,000 to 200,000 typical gates. The VHDL synthesis and simulation software used are Altera Max+Plus II 10.2 and LeonardoSpectrum Level 1.

3. Result and Discussion

Functional simulation, as the name implies, is a simulation to test its functionality only. During functional simulation, all the propagation delays between the logic gates are ignored. Therefore, the output logic levels change at the same time as the input vectors. Hence, perfect timing diagram can be obtained by functional simulation.

To start the functional simulation, the inputs of this module were fed in with the appropriate values for encryption and decryption. The simulation results were pasted below, which show the simulation of encryption and decryption.

By observing Figure 1 below, all the inputs are in hexadecimal, while the output "C" is in decimal format. Note that the input "N_C" is the complemented modulus. Once the "GO" signal was given, the calculation started. The output "C" was giving the final result when the output "DONE" changed to '1' (high state).

The Figure 2 below shows the decryption process to get back the original message, 7. The input "M" was fed with the cipher text from Figure 1 previously. The input "E" was changed to the private key, d = 29 (1DH) to carry out the decryption. Original message was obtained when the "DONE" = '1' (high state).

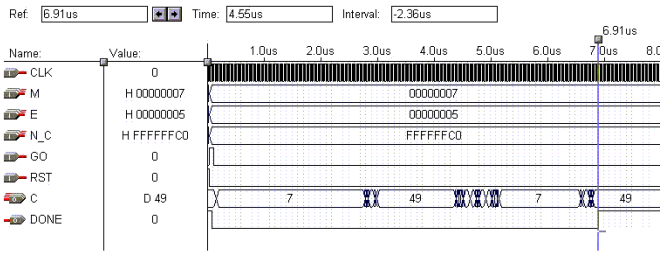


Figure 1. Encryption with the calculation of C = 75 mod 63.

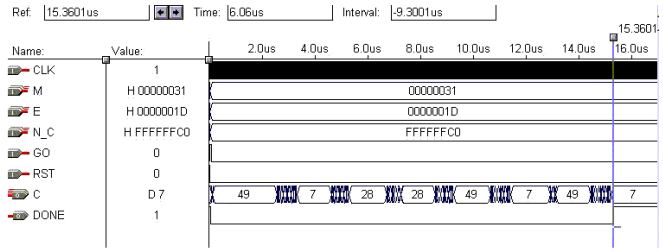


Figure 2. Decryption of M = 4929 mod 63.

From the synthesis results pasted above, the 3DES module was targeted on the Altera device, EPF10K30EQC208.

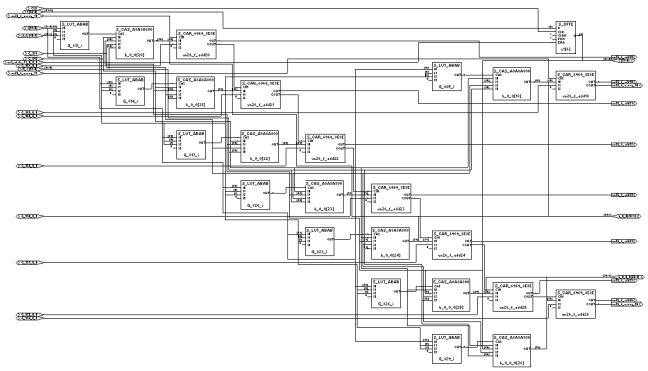
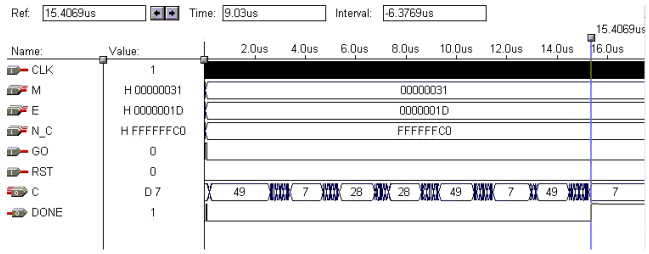


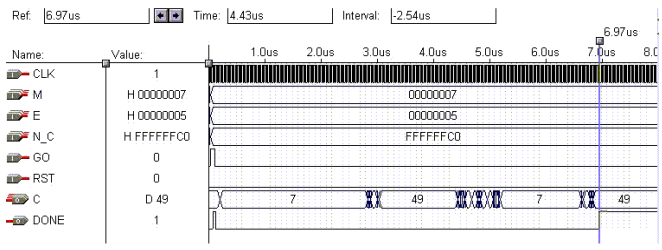
Figure 3. Technology view of 3DES module.

It has used up 239 units of logic cells, which is about 20% utilization of the chosen device. The frequency report shows that this module achieved a clock frequency of 199MHz.

After the process of logic synthesis, timing simulation will then allow us to verify the project before it is actually committed to the hardware. Timing simulation takes care of the propagation delay between logic gates, which represents the real timing information as if it is running in FPGA.



(a)



(b)

Figure 4. (a) Encryption. (b) Decryption with same set of parameters.

Table 5. S Box

S1															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S2															
15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S3															
10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S4															
7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S5															
2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S6															
12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S7															
4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S8															
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Figure 4a shows the encryption process while Figure 4b shows the decryption process, which used the same set of parameters.

Refer to Figure 4a and 4b, the identical results was obtained compare to Figure 1 and 2. However, the timing characteristics were different. The time taken for encryption and decryption process was 6.97us and 15.4069us respectively. This result can be compared to the functional simulation, where the propagation delay time is ignored. Referring back to Figure 4a and Figure 4b, the time taken was 6.91us and 15.3601us. Hence, we know that the propagation delay for encryption and decryption are 60ns and 46.8ns respectively.

Table 8. Comparison with the existing crypto engine

Encryption	Size	Speed (MHz)	Throughput (Mbit/s)
Triple DES	239 slices	199	796
DES	710 slices	168	668
AES Key Expander (32 bits datapath)	181 slices + 2 RAM Blocks	149	432.1
MD5	612	78.8	498.1
SHA-1	614 slices + 1 RAM Block	62	488.3

The performance of the hardware implementation of certain encryption mechanism varies with the technology used. As such, to compare different encryption method, it is essential to use the same FPGA architecture. Table 8 shows the result of comparison using Xilinx Virtex II. From the table it can be shown that the hardware realization of Triple DES is giving maximum throughput with higher frequency.

4. Conclusion

The primary goal of this research project was to develop a 3DES encryption engine with a sufficient security level. The 64-bit key size has made this particular 3DES encryption engine possible to achieve a significant level of security. Beside the security issue, another major concern of this research project was to achieve a faster processing time to accomplish the 64-bit 3DES operation. The 3DES encryption engine has been tested and verified to perform a 64-bit operation in less than 22.38us at 199 MHz clock speed. This result has given the sufficient ground to claim that the hardware realization of 3DES encryption engine, a faster one than other previous works. Most importantly, from the synthesis result, it has clearly been observed that the 3DES module has used only 239 units of total available LCs. This result resembles that the designed encryption engines takes a smaller space in the targeted FPGA and can be fitted in an FPGA with a smaller capacity.

5. References

1. Available from: http://www.iscit.surfnet.nl/team/Barry/thesis/crypto.html, 2015 Jan 9.
2. Aladdin Knowledge System, The Enduring Value of Symmetric Encryption. Aladdin Knowledge System (White Paper). 2000 Aug; 5–8.
3. Cheung OYH, Leong PHW. Implementation of an FPGA based accelerator for virtual private networks. Proceedings of IEEE International Conference on Field-Programmable Technology (ICFPT); 2002; Hong Kong. Berlin: Springer; 2002. p. 34–41.
4. Caloyannides MA. Encryption wars: early battles. IEEE Spectrum. 2000 Apr; 37(4):37–43.
5. Runje D, Kovac M. Universal strong encryption FPGA core implementation. Design, Automation and Test in Europe; 2013 Feb 23-26; 923–4 .
6. Kerins T, Popovici E, Daly A, Marnane WP. Hardware encryption engines for E-Commerce. Irish signals and Systems Conference; 2002 Jun. p.89–94.
7. Wu L, Weaver C, Austin T. Crypto maniac: A fast flexible architecture for secure communication. Proceeding of the 28th Annual International Symposium; 30 Jun – 4 Jul 2001; Goteborg, Sweden. 110–19.
8. Sanchez-Avilla C, Sanchez-Reillo R. The Rijndael block cipher (AES Proposal): A comparison with DES, Security Technology. IEEE 35th International Carnahan Conference; 2001 Oct 16-19. P. 229–34.
9. Chodowiec PR. Comparison of the hardware performance of the AES candidates using reconfigurable hardware [Master Thesis]. Fairfax, VA: George Mason University; 2002; 16–22.
10. Damgard IB, Knudsen LR. Two key triple encryption. Journal of Cryptology. 1998; 11(3):209–18.
11. Kaliski B. A survey of encryption standards. IEEE Micro. Dec 1993; 13(6):74–81.
12. Hamalainen P, Hannikainen M, Hamalainen T, Saarinen J. Configurable Hardware Implementation of Triple DES Encryption Algorithm for Wireless Local Area Network. Acoustics, Speech and Signal Processing (ICASSP'01). 2001; Tampere, Finland.1221–4.